1) آیا پایتون یک زبان مفسری است یا کامپایلری؟ توضیح دهید.

این زبان از زبان‌های برنامه‌نویسی مفسر بوده و به صورت کامل یک زبان شیءگرا است که در ویژگی‌ها با زبان‌های تفسیری پرل، روبی، اسکیم، اسمال‌تاک و تی‌سی‌ال مشابهت دارد و از مدیریت خودکار حافظه استفاده می‌کند

یک مفسر (interpreter) کد منبع را (همانگونه که برنامه توسط برنامه‌نویس نوشته شده) می‌خواند، تجزیه می‌کند و دستورالعمل (کد برنامه) را روی هوا تفسیر می‌کند. زمانی که ما از حالت تعاملی مفسر پایتون استفاده می‌کنیم، یک خط یا جمله‌ی پایتونی می‌نویسیم و پایتون در جا آن را خوانده، پردازش کرده و از ما طلب خط بعدی را می‌کند. البته این روند بستگی به میزان پردازش لازم هم دارد برای آن خط کد را نیز دارد.

برخی خطوط در پایتون، به او می‌گوید که «بایستی مقادیری را برای استفاده‌ی بعدی به خاطر بسپاری». برای این کار ما نامی برای آن مقدار برمی‌گزینیم که بتوانیم در آینده نیز آن نام را به خاطر آورده و مقدار متغیر را از آن خارج کنیم. اسم برچسب‌هایی که به این مقادیر می‌زنیم variable یا متغیر است. متغیرها، مقادیر را در خودشان نگه می‌دارند.

یک زبان برنامه‌نویسی «مفسری (Interpreted)» محسوب می‌شود. مفسر زبان پایتون دستورالعمل‌ها را خط به خط و بر اساس ترتیب آن‌ها (در مجموعه کدهای نوشته شده توسط برنامه‌نویس)، می‌خواند، ارزیابی می‌کند و خروجی‌های حاصل از اجرای دستورات را در خروجی نمایش می‌دهد. تمام این کارها توسط «مفسر (Interpreter)» انجام می‌شود.

2) آیا پایتون case sensitive است؟

در پایتون، کلیدواژه‌ها به حروف بزرگ و کوچک حساس هستند .(Case Sensitive) در واقع، نمی‌توان کلیدواژه‌ای که با حروف کوچک است را با حروف بزرگ نوشت یا بالعکس؛ در این صورت، برنامه دچار خطا می‌شود. ۳۳ کلیدواژه در پایتون ۳,۷ وجود دارد. این تعداد ممکن است در طول زمان و به تدریج شاهد تغییراتی باشد. در پایتون، همه کلیدواژه‌ها به جز False ، Trueو None ، با حروف کوچک هستند و باید درست به صورتی که هستند نوشته شوند. لیست همه کلیدواژه‌های زبان پایتون در ادامه ارائه شده است.

3) نحوهی نوشتن توضیحات در پایتون به چه شکل است؟

در پایتون، از علامت «هش (Hash)» یعنی # برای شروع نوشتن یک توضیح استفاده می‌شود. در صورتی که نیاز به نوشتن توضیحات در چند خط پشت سر هم باشد، باید در ابتدای هر خط از علامت # استفاده شود. راهکار دیگر، استفاده از سه علامت «نقل قول (Quotes)» انگلیسی به صورت ''' یا """ است. این علامت‌های نقل قول سه تایی به طور کلی برای

رشته‌های چند خطی مورد استفاده قرار می‌گیرند. در صورتی که توضیحات «داک‌استرینگ (Docstrings) «نباشد، هیچ کد اضافی ایجاد نخواهند کرد.

4) بلاک نویسی در پایتون به چه صورت انجام میگیرد؟

ابتدا ویرایشگر متن را باز کرده و یک فایل جدید با عنوان blockchain.py ساخته می‌شود. در ادامه همواره از همین یک فایل استفاده می‌شود. هر بلوک دارای یک «اندیس(index) «، »برچسب زمان (timestamp) «(به زمان یونیکس)، یک لیست از تراکنش‌ها، یک proof و هش بلوک قبلی است. بلاک چین (Blockchain) از دو کلمه Block(بلوک) و Chain (زنجیره) ایجاد شده است. این فناوری در حقیقت زنجیره‌ای از بلوک‌هاست.به طور کلی بلاک چین یک نوع سیستم ثبت اطلاعات و گزارش است.تفاوت آن با سیستم‌های دیگر این است که اطلاعات ذخیره شده روی این نوع سیستم، میان همه اعضای شبکه به اشتراک گذاشته می‌شوند و با استفاده از رمزنگاری امکان حذف و دستکاری اطلاعات ثبت شده تقریبا غیرممکن است.

5) تبدیل نوع متغیرها (type casting) در پایتون به چه صورتی انجام می شود . توضیح دهید.

پایتون دارای دو نوع تبدیل نوع است.

- تبدیل نوع ضمنی(Implicit Type Conversion)
- تبدیل نوع صریح(Explicit Type Conversion)

تبدیل نوع ضمنی : در تبدیل نوع ضمنی، پایتون به طور خودکار یک نوع داده را به نوع دیگری تبدیل می‌کند. این فرایند نیازی به درگیر کردن کاربر ندارد.

تبدیل نوع داده صریح : در تبدیل نوع داده صریح، کاربر نوع داده یک شی را به یک نوع داده مورد نیاز تبدیل می‌کند. برای انجام تبدیل نوع صریح، از توابع از پیش تعریف شده‌ای مانند ()str() ، floatو ()int استفاده می‌شود. این تبدیل نوع، typecasting نیز نامیده می‌شود، زیرا کاربر نوع داده شی را تغییر می‌دهد.

6) منظور از انقیاد نوع پویا (late binding) در پایتون چیست؟ آیا نوع متغیر در طول روند اجرای یک برنامه تغییر میکند؟

به انقیاد صورت گرفته در زمان اجرای برنامه ، انقیاد دیررس (late binding) یا پویا (dynamic bidning) یا مجازی (virtual binding) می گویند .به عنوان نمونه انقیاد نام یک تابع به صورت زودرس صورت می‌گیرد؛ تابع انقیاد شده به یک نام نمی‌تواند در زمان اجرا تغییر کند.

7) تفاوت لیست در پایتون با آرایه در سایر زبان های برنامه نویسی در چیست؟ همراه با ذکر مثال توضیح دهید.

لیست یک مجموعه از داده ها که قابل تغییر و مرتب است و می تواند اعضای تکراری داشته باشد.

```
Thislist =[ " apple"  ,"banana"  ,"cherry"]
Print ( thislist )
```

در برنامه‌نویسی پایتون، نوع داده لیست با قرار دادن همه آیتم‌ها (عناصر) درون یک براکت (کمانک) یعنی [ ]، ساخته می‌شود. عناصر یک لیست، با استفاده از علامت ویرگول، یعنی ","، از هم جدا می‌شوند. لیست می‌تواند هر تعدادی عنصر داشته باشد و این عناصر ممکن است خود دارای انواع داده متفاوتی باشند. برای مثال، عناصر یک لیست ممکن است ترکیبی از نوع داده صحیح(integer) ، شناور (float) و رشته (String) باشند .

آرایه ها (Array) در واقع متغیر های خاصی هستند که می تونند چندین داده را در خود ذخیره کنند.

```
Cars = ["Ford" ,"Volvo" ,"BMW"]
```

برای دسترسی به آرایه ها از نام آرایه اندیس عنصر استفاده می شود

8) تفاوت نوع دادهی tuple و list در پایتون چیست؟ با ذکر مثال بیان کنید.

تفاوت کلیدی بین لیست و تاپل در این است یک لیست قابل تغییر است. پس از ایجاد می توان آن را تغییر داد ولی تاپل غیرقابل تغییر است. پس از ایجاد امکان تغییر آن وجود ندارد.

لیست یک نوع داده مرکب به زبان برنامه نویسی پایتون است که می تواند انواع مختلفی از داده ها را ذخیره کند و می تواند عناصر را پس از ایجاد تغییر دهد ولی tuple یک نوع داده مرکب به زبان برنامه نویسی پایتون است که می تواند انواع مختلفی از داده ها را ذخیره کند و نمی تواند عناصر را پس از ایجاد تغییر دهد.

عناصر یک لیست در براکت های مربع محصور شده اند ولی عناصر یک تاپل در پرانتز محصور شده است.

تکرار از طریق عناصر در یک لیست به سرعت در یک tuple انجام نمی شود ولی حرکت از طریق عناصر در یک tuple سریعتر از لیست است.

9) – نوع دادهی dictionary را به همراه یک مثال توضیح دهید. دسترسی توسط کلید به صورت انجام میگیرد؟

نوع داده دیکشنری (Dictionary) در زبان برنامه نویسی پایتون(Python) ، به صورت لیستی از، کلیدها و value ها است، هر کلید توسط علامت کالن (:) از value جدا می شود، اندیس ها نیز توسط علامت کاما (,) از یکدیگر جدا می شوند. یک دیکشنری خالی بدون اندیس با استفاده از تنها دو آکولاد خالی به صورت {} نوشته می شود.

کلید ها در نوع داده dictionary یکتا می باشد (بدین معنی که نمی توان دو کلید یکسان داشت)، ولی value می تواند تکراری نیز باشد .value در دیکشنری می تواند هر نوعی باشد، ولی کلیدها باید یک نوع داده غیر قابل تغییر (immutable)مانندstring ، number، یا tuble باشد.

مثال:

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print ("dict['Name']: ", dict['Name'])
print ("dict['Age']: ", dict['Age'])
```

خروجی :

```
dict['Name']:  Zara
dict['Age']:  7
```

10) مفهوم negative indexing در list به چه معناست؟

چاپ از آخر لیست انجام شود

مثال :

```
b = "Hello, World!"
print(b[-5:-2])
```

خروجی: orl

11) در صورتی که بخواهیم عناصر تکراری یک لست را حذف کنیم، راهحل خود را پیشنهاد دهید.

چهار روش برای حذف عناصر تکراری پیشنهاد میشه

1)–نوشتن برنامه با استفاده از دیکشنری پایتون 2)– نوشتن یک تابع

3)–استفاده از دستورnot 4)–استفاده از دستور set

12) کاربرد نوع دادهی Set را بیان کنید.

کاربرد نوع داده Python Set حذف داده های تکراری است. در این حالت از قطعه کد زیر استفاده می‌شود.

```
GoT = ['Ali', 'Reza', 'Iman', 'Mojtaba', 'Iman', 'Mojtaba', 'Ali']
finalData = list(set(GoT))
print(finalData)
```

13) چگونگی افزودن یک عنصر داده یا یک لیست از داده ها به یک لیست پایتونی را با مثال توضیح دهید.

می‌توان «یک عنصر» را با استفاده از متد append() و «چند عنصر» را با استفاده از متد extend() به لیست اضافه کرد.

```
odd = [1, 3, 5]
```

```
odd.append(7)
 # Output: [1, 3, 5, 7]
print(odd)
odd.extend([9, 11, 13])
 # Output: [1, 3, 5, 7, 9, 11, 13]
print(odd)
```

14) چگونگی دسترسی به عناصر یک tuple را بیان کنید. آیا میتوان عناصر tuple را تغییر داد؟

به یک عنصر خاص در تاپل با استفاده از ایندکس ها دسترسی پیدا کنید. نکته قابل توجه این است که در پایتون از اعداد منفی برای دسترسی به عناصر از آخر و به صورت معکوس استفاده میکنیم. به عنوان مثال عدد ۱- به آخرین عنصر مجموعه اشاره دارد. مطمئناً اگر تلاش کنیم با یک ایندکس نامعتبر به عنصری خارج از محدوده تاپل دسترسی پیدا کنیم، با خطای IndexErrorرو به رو میشویم

مفهوم دیگری که اغلب در استفاده از تاپل ها شنیده اید، unpackingاست که می تواند دسترسی به عناصر مشخصی را برای شما فراهم آورد .

یک تاپل دنباله ای تغییر ناپذیر از مقادیر است. بنابراین نمیتوانیم مقادیر آن را تغییر دهیم.

اگرچه یک تاپل به عنوان یک شیء در کل قابل تغییر نیست، اما اگر خود عناصر قابل تغییر باشند، میتوانیم آن مقدار مشخص را تغییر داد.

15) کاربرد متدهای split , trim, upper, lower را در پردازش رشتهها توضیح دهید.

Split() : بوسیله این تابع می توان یک رشته را بر اساس یک کاراکتر مشخص به تعدادی رشته تقسیم کرد. خروجی این تابع یک لیست است .

متد upper() : کلیه کاراکترهای رشته را تبدیل به حروف بزرگ می کند

متد Lower() : کلیه کاراکترهای رشته را تبدیل به حروف کوچک می کند

متد trim() : کاراکترهای space (فاصله) را از ابتدا و انتهای رشته ی مورد نظر حذف می کند.

16) ساختار شرط در پایتون به چه صورتی است، آیا میتوان دستورات بلاک شرط را ماقبل شرط قرار داد؟

یک دستور شرطی ساده با کیورد ifبه معنای «اگر» آغاز می‌شود و مفسر پایتون با رسیدن به این کیورد تصمیم‌گیری می‌کند که از همین روی در ادامهٔ دستور ifبه دنبال شرط مد نظر می‌گردد. در ادامه و پس از نوشتن شرط مورد نظر علامت :درج می‌گردد

و بدین ترتیب هِدرِ این دستور مرکب به پایان می‌رسد. حال نوبت به بدنهٔ دستور if می‌رسد که همچون هر دستور مرکب دیگری با رعایت تورفتگی نسبت به بلوک هِدرِ نوشته می‌شود و در صورتی که شرط if برقرار باشد دستورات داخل بدنه اجرا می‌شوند. خیر دستورات شرط بعد از شرط اجرا می شوند

17) کاربرد تابع zip و enumerate را در حلقه بنویسید

enumerate() : یک تابع از پیش‌ساخته شده و موجود در خود برنامه (Built-in) در پایتون (python) است! تابع enumerate برای یک لیست، عناصر و اندیس‌های آن را با هم در نظر می‌گیرد!

هنگامی که در یک دنباله از حلقه استفاده شود، با استفاده از تابع enumerate() میتوان اندیس مکان و مقدار متناظر با آن را به صورت همزمان بازیابی کرد.

zip : برای آنکه بتوان عملیات درون حلقه را به طور همزمان بر روی بیش از یک توالی اجرا کرد، از تابع zip استفاده می شود.

تابع zip در پایتون برای متناظر کردن چندین لیست و تاپل بکار میرود

18) چگونگی تبدیل نوع دادهی datetime به رشته و بلعکس را توضیح دهید.

مثال : برای تبدیل timestamp به datetime

```
from datetime import datetime

timestamp = 1545730073
dt_object = datetime.fromtimestamp(timestamp)

print("dt_object =", dt_object)
print("type(dt_object) =", type(dt_object))
```

در اینجا، کلاس datetime از ماژول» datetime وارد) «ایمپورت (Import | شده است. سپس، از Classmethod با عنوان datetime.fromtimestamp() استفاده است؛ این Classmethod ، تاریخ و زمان محلی) شی (datetime را باز می‌گرداند. شی در متغیر dt_object ذخیره شده است.

مثال : برای تبدیل datetime به timestamp

```
from datetime import datetime

# current date and time
```

```
now = datetime.now()

timestamp = datetime.timestamp(now)
print("timestamp =", timestamp)
```

19) تعیین مقدار پیشفرض برای پارامترهای یک تابع به چه نحوی انجام میگیرد؟ توضیح دهید در چه صورت میتوان تابعی را بدون پارامترهایش فراخوانی کرد؟

```
def main(arg1 , arg 2 ='ali')
```

در صورتی که در زمان تعریف یک تابع برای پارامترهای آن مقدار پیش فرض تعیین کرده باشیم

20) تابع لامبدا چیست؟ با مثال توضیح دهید.

لامبدا یک روش ساده برای تعریف تابع در پایتون است. این توابع غالباً به نام «عملگرهای لامبدا» یا «تابع‌های لامبدا» نامیده می‌شوند.

مثال :

```
def add_five(number):
return number + 5
print(add_five(number=4))
```

تابع لامبدای معادل آن چنین است:

```
add_five = lambda number: number + 5
print(add_five(number=4))
```

21) کاربرد دو تابع map و filter را در لامبدا نویسی با مثال بیان کنید.

تابع map در پایتون این امکان را به شما می‌دهد که یک تابع را روی تمام اعضای یک لیست اعمال کنید.

نحوه تعریف map به صورت زیر است.

```
Map(function , list_name)
```

ورودی اول دستور map یک تابع است که باید روی لیست اعمال شود که معمولا یک lambda function است و ورودی دوم دستور نام لیستی است که تابع روی ان اعمال می‌شود.

```
map(lambda x: x**2, items)
```

قابلیت filter شبیه به مپ عمل می‌کند با این تفاوت که امکان چک کردن یک شرط را روی تمام اعضای یک لیست را فراهم می‌کند.

```
filter(lambda x: x < 0, number_list)
```

همانطور که مشاهده می‌کنید فیلترها همانند مپ ها دو ورودی دارند. ورودی اول یک lambda function است که شرط مورد نظر داخل آن تعریف می‌شود و ورودی دوم نام لیستی است که این شرط روی آن اعمال می‌شود. خروجی این دستور لیستی از آیتم هاست که مقدار تابع در این آیتم ها true بوده است.

22)استثنا چیست؟ مکانیزم استثنا گردانی در پایتون به چه نحوی است؟

استثناها مشکلاتی هستند که در هنگام اجرای برنامه‌ها ممکن است رخ دهند

طبیعی است استثناهایی که در برنامه ممکن است رخ دهد، باید مدیریت شوند تا در کارکرد بقیه اجزای برنامه مشکل بوجود نیاورد.

قدم اول این است کدی که حدس می‌زنیم به مشکل برخورد کند را در try بگذاریم. قدم بعدی استفاده از except برای مدیریت کردنِ استثنا، در صورتی که کدی که در try قرار دارد به مشکل برخورد کند، است.

زمانی که کد درونِ try اجرا می‌شود، اگر به مشکلی برخورد نکند except را نادیده می‌گیرد و برنامه به کار خود ادامه می‌دهد. اما اگر به مشکل برخورد کند، برنامه ادامه کد را اجرا نمی‌کند و به except می‌رود. اگر نامِ استثنا با استثنایی که رخ داد یکسان بود کد درونِ exceptاجرا می‌شود، در غیر این صورت برنامه با error متوقف می‌شود.